

Fig. 2A

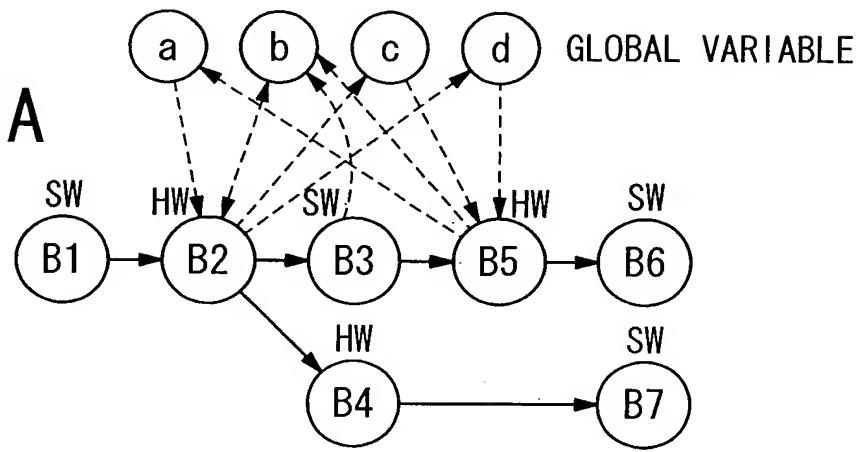


Fig. 2B

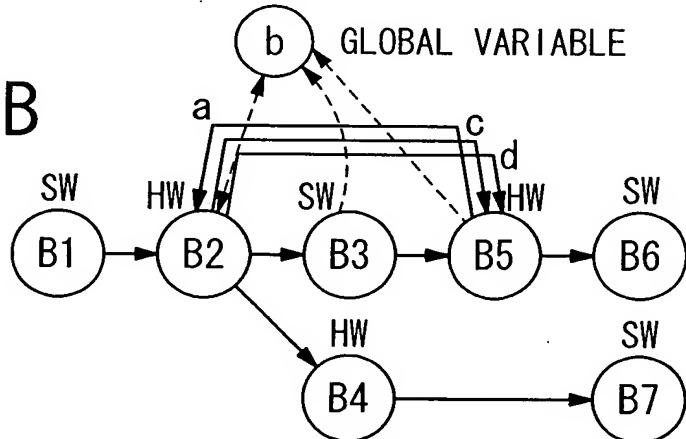


Fig. 3A

```

int a, b, c, d;

funcB2 (in1, *out1) {
    a:read
    b = a+b+in1;
    c = a+1;
    d = a+2;
    out1 = c+a;
}

funcB3 (in2) {
    b:read
    b = in2+b;
}

```

3B
b6
F

int b;

```

funcB2(in1, *out1) {
    a:read
    b:read, write
    c:write
    d:write
    out1 = c+a;
}

funcB3(in2) {
    b = in2+b;
}

funcB5(in3) {
    a = in3+c+d;
    b = c+d;
}

funcB2(in1, a, *out1, *c, *d) {
    b=a+b+in1;
    c=a+1;
    d=a+2;
    out1=c+a;
}

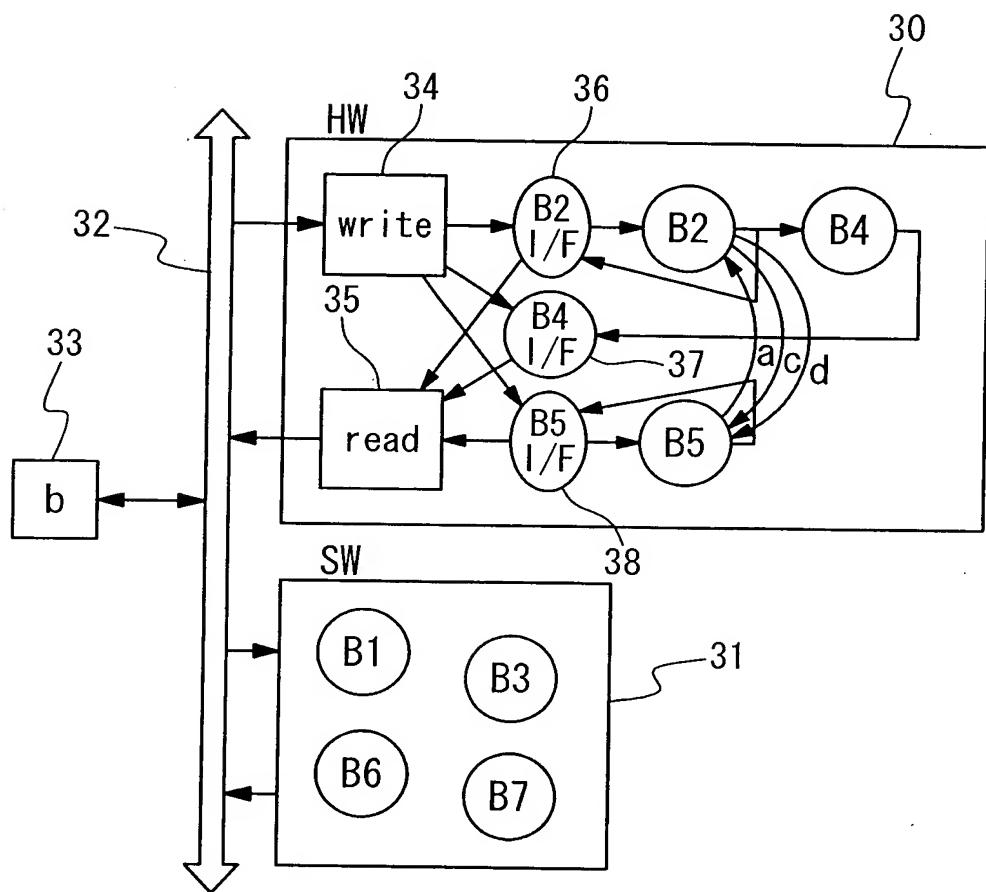
funcB3(in2) {
    b=in2+b;
}

funcB5(in3, *a, c, d) {
    a=in3+c+d;
    b=c+d;
}

```

3C
b
F

Fig. 4



F i g . 5

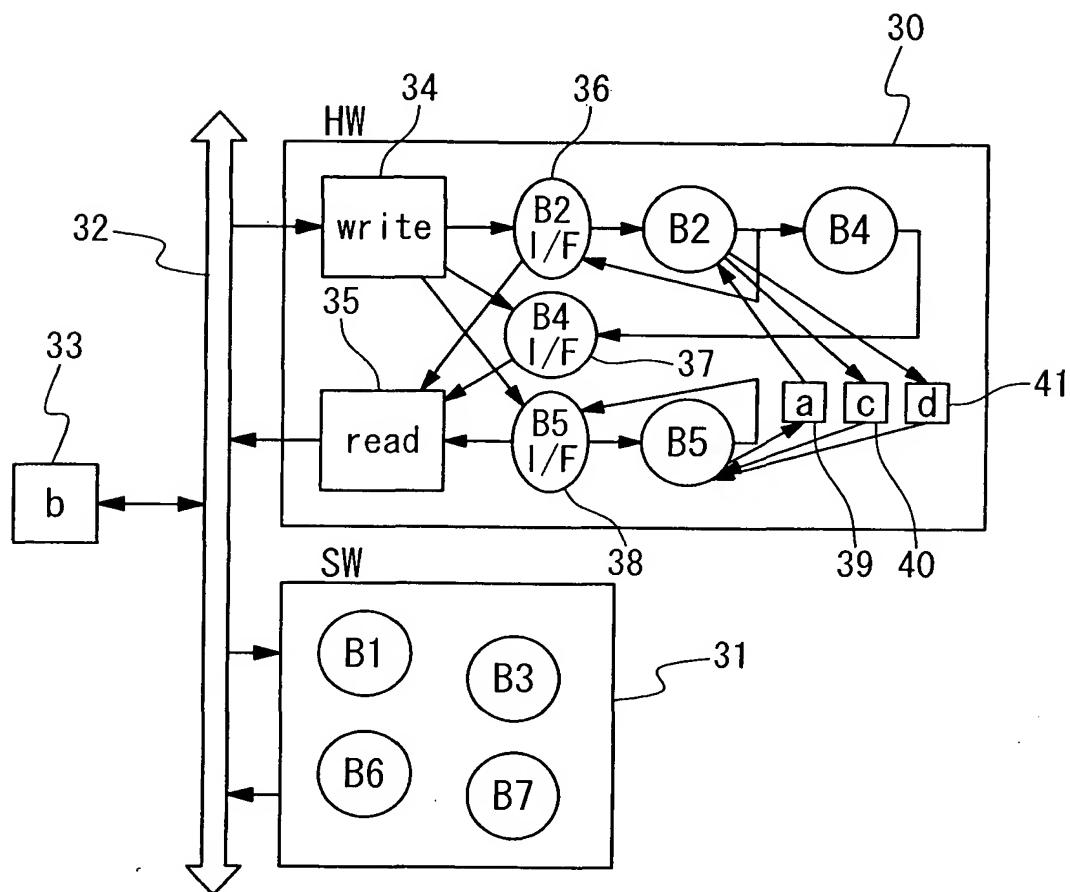


Fig. 6

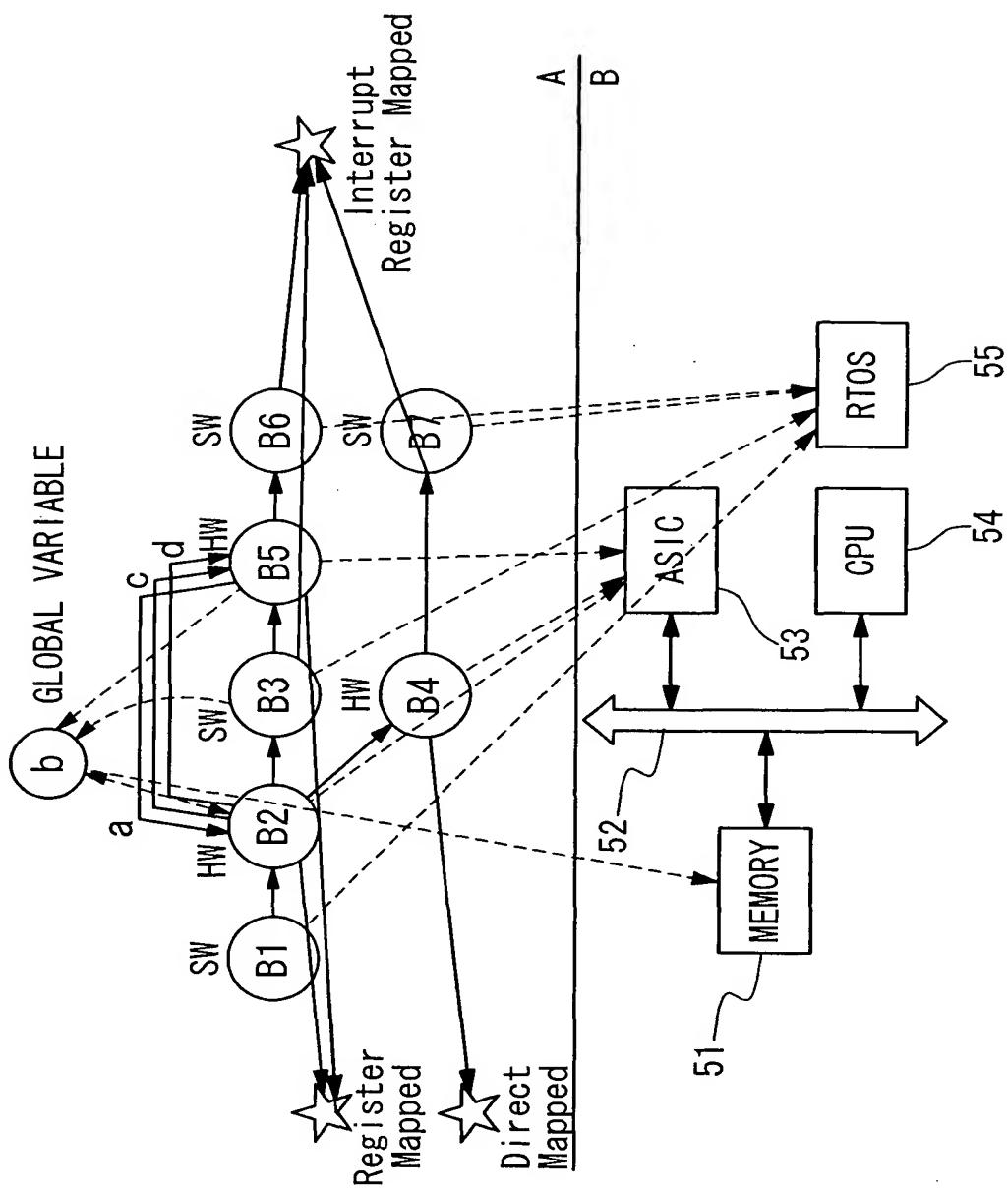


Fig. 7

```
BUS bus_1("Bus1", arbitor_1, busread_1, buswrite_1);
Int arbitor_1(void);
RDATA busread_1(int add, int byte_count);
Int buswrite_1(int add, int data, int byte_count);

CPU ("cpu_1", &cpu_1_busif, 0, 0x10000, 0x00ffe000, 0x1000, 0x8000);
BusIntf cpu_1_busif("cpu 1 Bus I/F", &bus_1, 1)

SDRAM ExtSDRAM1("External SDRAM 1", EXTSDRAM1_START_ADD, EXTSDRAM1_SIZE)

c_cpuifB2 cpuifB2(); c_cpuifB4 cpuifB4(); c_cpuifB5 cpuifB5();
c_B2 funcB2(); c_B4 funcB4(); c_B5 funcB5();

SystemoneScep()
{
    funcB2.OneStep(); cpuifB2.OneStep();
    funcB4.OneStep(); cpuifB4.OneStep();
    funcB5.OneStep(); cpuifB5.OneStep();
}
Systeminit()
{
    funcB2.init(); cpuifB2.int();
    funcB4.init(); cpuifB4.int();
    funcB5.init(); cpuifB5.int();
}
SystemReset()
{
    funcB2.Reset(); cpuifB2.Reset();
    funcB4.Reset(); cpuifB4.Reset();
    funcB5.Reset(); cpuifB5.Reset();
}
```

Fig. 8

```
RDATA busread_1 (int add, int byte_count) {
    if ((add < start_bus_address) || (add > end_bus_address)) {
        return(NULL);
    }
    //RAM
    if ((add >= ExternalRAM.StartAddress) && (add < ExternalRAM.EndAddress)) {
        return ExternalRAM.Read(add - ExternalRAM.StartAddress, byte_count);
    }

    //ASIC
    if ((start_sub_address+B2_start_offset <= add)
        && (add < start_sub_address+B2_end_offset)) {
        cpuifB2. cpuread = 0;
        cpuifB2. bus_addin = add;
        rd.Status = cpuifB2.ReadIOReg();
        rd.data = cpuifB2.dataout;
    }
    if ((start_sub_address+B4_start_offset <= add)
        && (add < start_sub_address+B4_end_offset)) {
        cpuifB4. cpuread = 0;
        cpuifB4. bus_addin = add;
        rd.Status = cpuifB4.ReadIOReg();
        rd.data = cpuifB4.dataout;
    }
    if ((start_sub_address+B5_start_offset <= add)
        && (add < start_sub_address+B5_end_offset)) {
        cpuifB5. cpuread = 0;
        cpuifB5. bus_addin = add;
        rd.Status = cpuifB5.ReadIOReg();
        rd.data = cpuifB5.dataout;
    }

    -----
    return (rd);
}
```

Fig. 9

```
RDATA buswrite_1 (int add, int byte_count)  {
    if ((add < start_bus_address) || (add > end_bus_address))  {
        return(1);
    }
    //RAM
    if ((add >= ExternalRAM.StartAddress) && (add < ExternalRAM.EndAddress))  {
        return ExternalRAM.Write(add - ExternalRAM.StartAddress, byte_count);
    }

    //ASIC
    if ((start_sub_address+B2_start_offset <= add)
        && (add < start_sub_address+B2_end_offset))  {
        cpuifB2. cpuwrite = 1;
        cpuifB2. bus_addin = add;
        cpuifB2. datain = data;
        rd.Status = cpuifB2.WriteIOReg();
    }
    if ((start_sub_address+B4_start_offset <= add)
        && (add < start_sub_address+B4_end_offset))  {
        cpuifB4. cpuwrite = 1;
        cpuifB4. bus_addin = add;
        cpuifB4. datain = data;
        rd.Status = cpuifB4.WriteIOReg();
    }
    if ((start_sub_address+B5_start_offset <= add)
        && (add < start_sub_address+B5_ens_offset))  {
        cpuifB5. cpuwrite = 1;
        cpuifB5. bus_addin = add;
        cpuifB5. datain = data;
        rd.Status = cpuifB5.WriteIOReg();
    }

    -----
    return (0);
}
```

Fig. 10

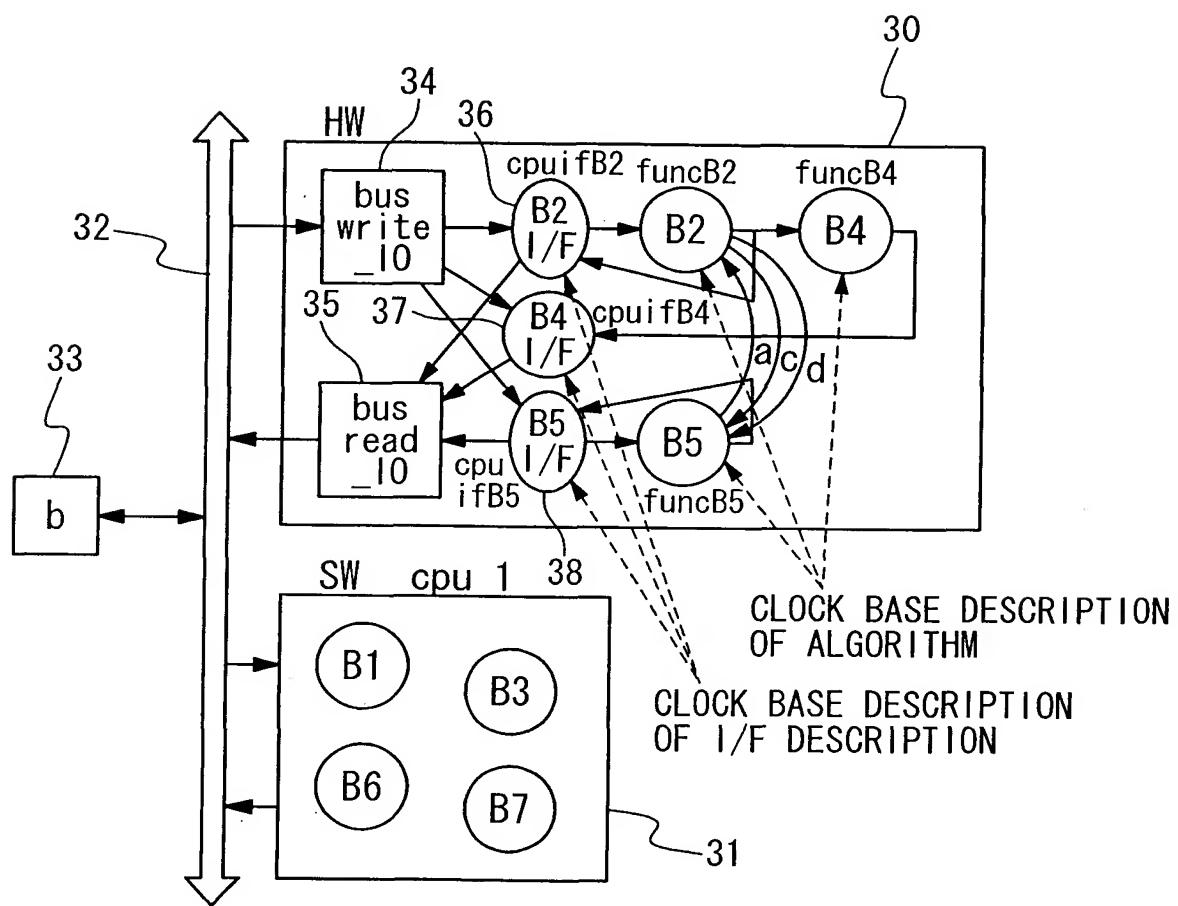


Fig. 11

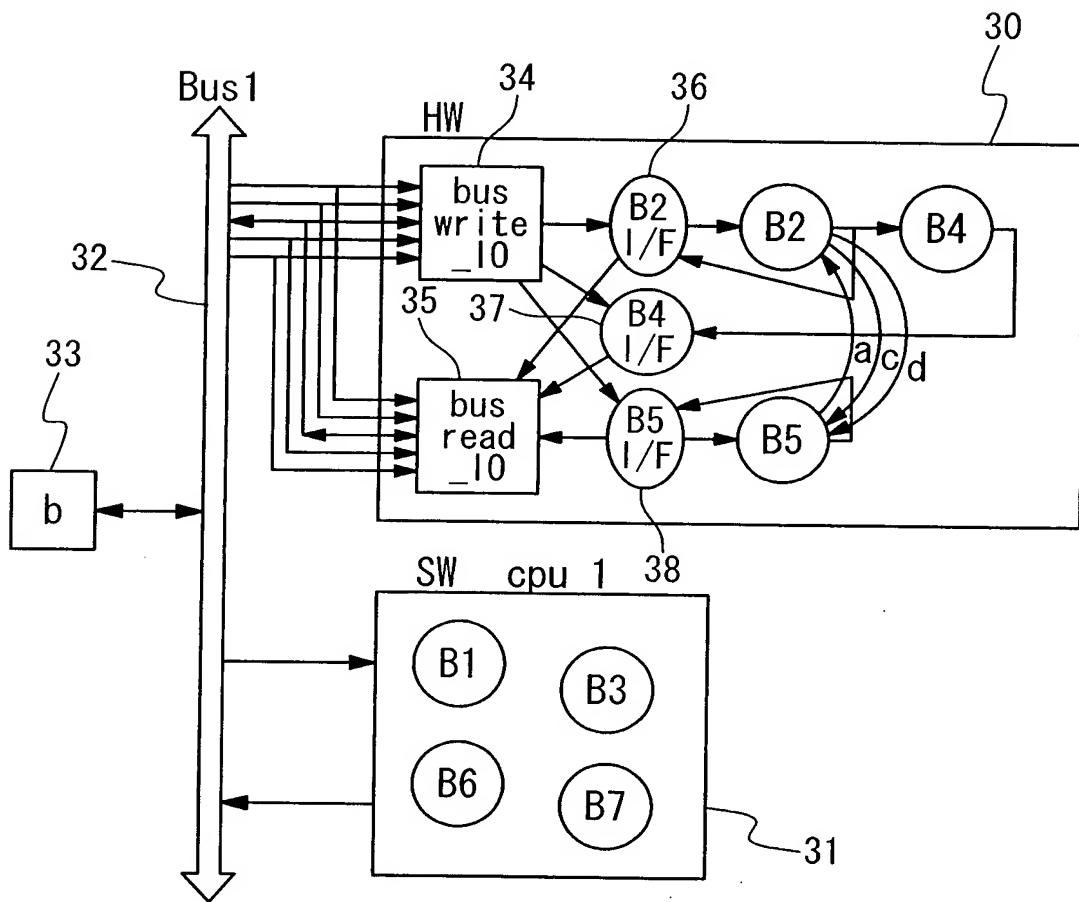


Fig. 12

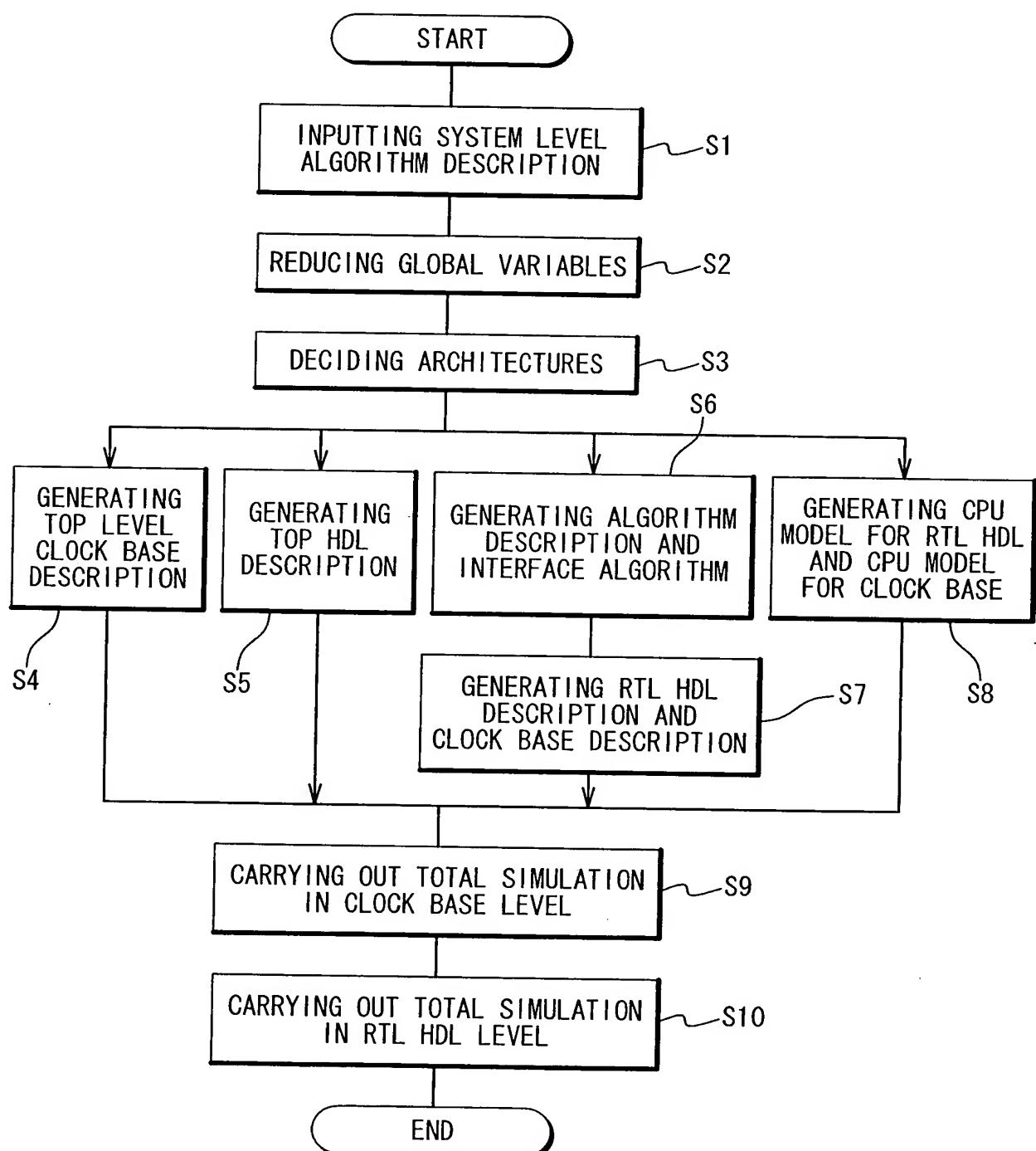


Fig. 13

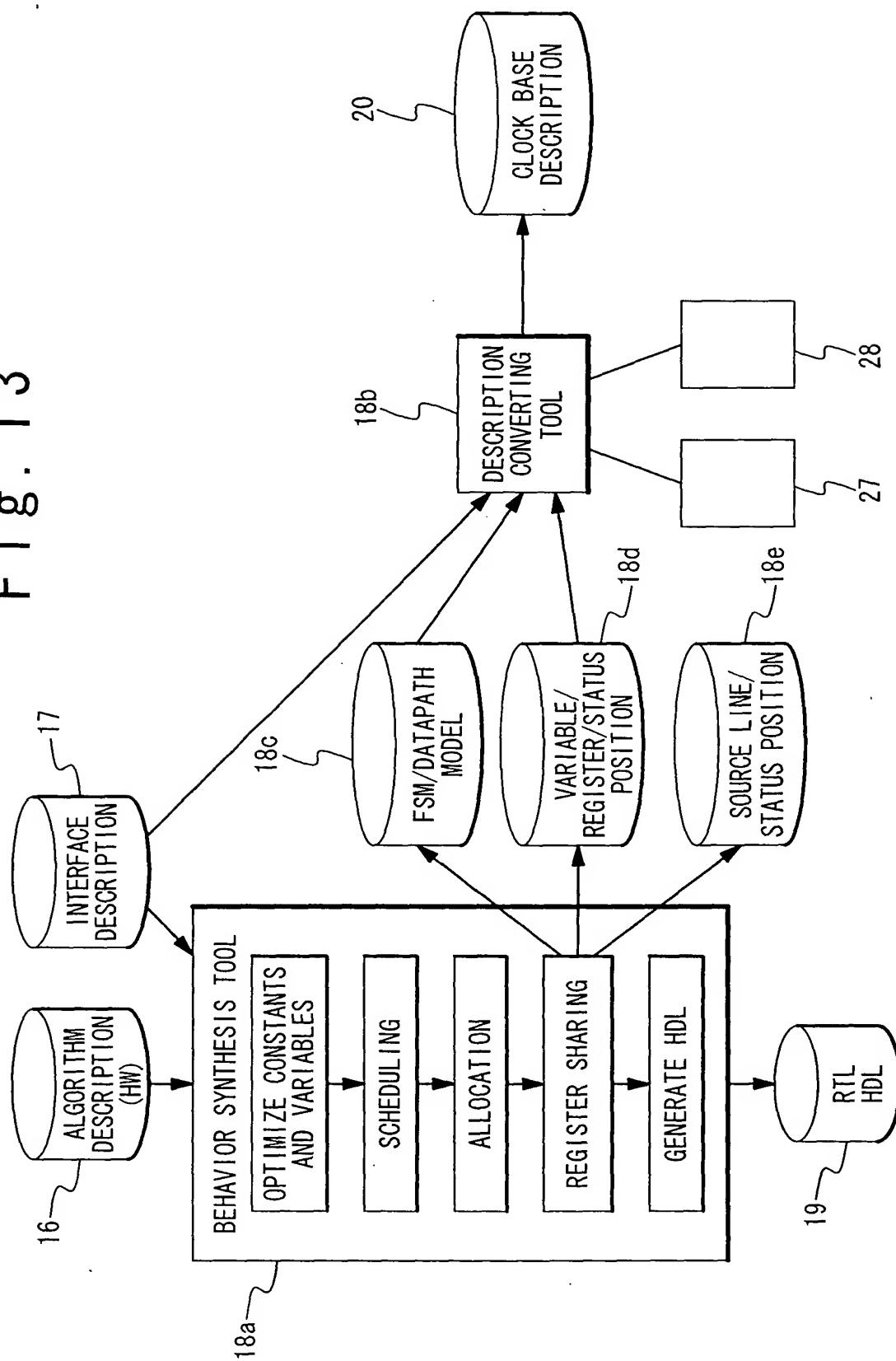


Fig. 14

